**NAME: GWAM ONYELUKACHUKWU NORA**

**MATRICULATION: ENG1508326**

**COURSE: CPE 321**

**LEVEL: 300**

**QUESTIONS**

**1.Discuss features of programming languages.**

**2.Outline 3 distinct programming languages per category.**

**3.In a tabular form,group the programming languages according to type.**

**4.Write simple/basic examples of programming languages and run a simple program for all languages.**

**5.compare Java and C# and state their uses.**

**COMMERCIAL PROGRAMMING LANGUAGE**

**List of programming languages by type**

1. **Array language.**

* **[A+](https://en.wikipedia.org/wiki/A%2B_%28programming_language%29" \o "A+ (programming language))**
* **[Analytica](https://en.wikipedia.org/wiki/Analytica_%28software%29" \o "Analytica (software))**
* **[APL](https://en.wikipedia.org/wiki/APL_%28programming_language%29" \o "APL (programming language))**
* **[Chapel](https://en.wikipedia.org/wiki/Chapel_%28programming_language%29" \o "Chapel (programming language))**
* **[Fortran](https://en.wikipedia.org/wiki/Fortran" \o "Fortran)**
* **[Freemat](https://en.wikipedia.org/wiki/Freemat" \o "Freemat)**
* **[GAUSS](https://en.wikipedia.org/wiki/GAUSS_%28software%29" \o "GAUSS (software))**
* **[J](https://en.wikipedia.org/wiki/J_%28programming_language%29" \o "J (programming language))**
* **[Julia](https://en.wikipedia.org/wiki/Julia_%28programming_language%29" \o "Julia (programming language))**
* **[K](https://en.wikipedia.org/wiki/K_%28programming_language%29" \o "K (programming language))**
* **[MATLAB](https://en.wikipedia.org/wiki/MATLAB" \o "MATLAB)**

1. **Authoring languages**

* **[Bigwig](https://en.wikipedia.org/w/index.php?title=Bigwig_%28programming_language%29&action=edit&redlink=1" \o "Bigwig (programming language) (page does not exist)) (web design and development language)**
* **[PILOT](https://en.wikipedia.org/wiki/PILOT" \o "PILOT)**
* **[TUTOR](https://en.wikipedia.org/wiki/TUTOR_%28programming_language%29" \o "TUTOR (programming language))**
* **[Lasso](https://en.wikipedia.org/wiki/Lasso_%28programming_language%29" \o "Lasso (programming language))**

1. **Command line interface languages**

**[Command-line interface](https://en.wikipedia.org/wiki/Command-line_interface" \o "Command-line interface) (CLI) languages are also called batch languages, or job control languages. Examples:**

* **[4DOS](https://en.wikipedia.org/wiki/4DOS" \o "4DOS) (extended command-line shell for [IBM PCs](https://en.wikipedia.org/wiki/IBM_Personal_Computer" \o "IBM Personal Computer))**
* **[bash](https://en.wikipedia.org/wiki/Bash_%28Unix_shell%29" \o "Bash (Unix shell)) (the Bourne-Again shell from [GNU](https://en.wikipedia.org/wiki/GNU" \o "GNU)/[FSF](https://en.wikipedia.org/wiki/Free_Software_Foundation" \o "Free Software Foundation))**
* **[CHAIN](https://en.wikipedia.org/w/index.php?title=CHAIN_%28programming_language%29&action=edit&redlink=1" \o "CHAIN (programming language) (page does not exist)) ([Datapoint](https://en.wikipedia.org/wiki/Datapoint" \o "Datapoint))**
* **[CLIST](https://en.wikipedia.org/wiki/CLIST" \o "CLIST) ([MVS](https://en.wikipedia.org/wiki/MVS" \o "MVS) Command List)**
* **[CMS EXEC](https://en.wikipedia.org/wiki/CMS_EXEC" \o "CMS EXEC)**
* **[csh](https://en.wikipedia.org/wiki/C_shell" \o "C shell) and [tcsh](https://en.wikipedia.org/wiki/Tcsh" \o "Tcsh) (C-like shell from [Bill Joy](https://en.wikipedia.org/wiki/Bill_Joy" \o "Bill Joy) at UC [Berkeley](https://en.wikipedia.org/wiki/BSD" \o "BSD))**
* **[DCL](https://en.wikipedia.org/wiki/DIGITAL_Command_Language" \o "DIGITAL Command Language) DIGITAL Command Language – standard CLI language for [VMS](https://en.wikipedia.org/wiki/OpenVMS" \o "OpenVMS) ([DEC](https://en.wikipedia.org/wiki/Digital_Equipment_Corporation" \o "Digital Equipment Corporation), [Compaq](https://en.wikipedia.org/wiki/Compaq" \o "Compaq), [HP](https://en.wikipedia.org/wiki/Hewlett-Packard" \o "Hewlett-Packard))**
* **[DOS batch language](https://en.wikipedia.org/wiki/Batch_file" \o "Batch file) (standard CLI/batch language for the [IBM PC](https://en.wikipedia.org/wiki/IBM_Personal_Computer" \o "IBM Personal Computer) running [DOS](https://en.wikipedia.org/wiki/DOS" \o "DOS) operating systems, popular before [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows" \o "Microsoft Windows))**

1. **Compiled languages**

* **[Ada](https://en.wikipedia.org/wiki/Ada_%28programming_language%29" \o "Ada (programming language)) (multi-purpose language)**
* **[ALGOL](https://en.wikipedia.org/wiki/ALGOL" \o "ALGOL) (extremely influential language design – the second high level language compiler)** 
  + **[SMALL Machine Algol Like Language](https://en.wikipedia.org/wiki/SMALL" \o "SMALL)**
* **[Ateji PX](https://en.wikipedia.org/wiki/Ateji_PX" \o "Ateji PX), an extension of the Java language for parallelism**
* **[BASIC](https://en.wikipedia.org/wiki/BASIC" \o "BASIC) (some dialects, including the first version of Dartmouth BASIC)**
* **[BCPL](https://en.wikipedia.org/wiki/BCPL" \o "BCPL)**
* **[Blue](https://en.wikipedia.org/wiki/Blue_%28programming_language%29" \o "Blue (programming language))**
* **[C](https://en.wikipedia.org/wiki/C_%28programming_language%29" \o "C (programming language)) (one of the most widely used procedural programming languages)**
* **[C++](https://en.wikipedia.org/wiki/C%2B%2B" \o "C++) (One of the most widely used object-oriented (OO) languages specially used in large-scale, highly complex, high-performance software systems)**
* **[C#](https://en.wikipedia.org/wiki/C_Sharp_%28programming_language%29" \o "C Sharp (programming language)) (compiled into [intermediate language](https://en.wikipedia.org/wiki/Intermediate_language" \o "Intermediate language), which generates a native image at runtime)**
* **[Visual Basic](https://en.wikipedia.org/wiki/Visual_Basic" \o "Visual Basic) (Earlier versions compiled directly to a native runtime. Recent .NET versions compile into intermediate language that is just-in-time compiled into a native image at runtime.)**
* **[Visual FoxPro](https://en.wikipedia.org/wiki/Visual_FoxPro" \o "Visual FoxPro)**
* **[Visual Prolog](https://en.wikipedia.org/wiki/Visual_Prolog" \o "Visual Prolog)**
* **[X++](https://en.wikipedia.org/wiki/Microsoft_Dynamics_AX" \o "Microsoft Dynamics AX)**
* **[X#](https://en.wikipedia.org/wiki/X_Sharp_%28programming_language%29" \o "X Sharp (programming language))**
* **[XL](https://en.wikipedia.org/wiki/XL_%28programming_language%29" \o "XL (programming language))**
* **[Z++](https://en.wikipedia.org/wiki/Z%2B%2B" \o "Z++)**

1. **Concurrent languages**

* **[Ada](https://en.wikipedia.org/wiki/Ada_%28programming_language%29" \o "Ada (programming language)) (multi-purpose language)**
* **[Alef](https://en.wikipedia.org/wiki/Alef_%28programming_language%29" \o "Alef (programming language)) – concurrent language with threads and message passing, used for systems programming in early versions of [Plan 9 from Bell Labs](https://en.wikipedia.org/wiki/Plan_9_from_Bell_Labs" \o "Plan 9 from Bell Labs)**
* **[Ateji PX](https://en.wikipedia.org/wiki/Ateji_PX" \o "Ateji PX) an extension of the Java language for parallelism**
* **[ChucK](https://en.wikipedia.org/wiki/ChucK" \o "ChucK) – domain specific programming language for audio, precise control over concurrency and timing**
* **[Cilk](https://en.wikipedia.org/wiki/Cilk" \o "Cilk) – a concurrent [C](https://en.wikipedia.org/wiki/C_%28programming_language%29" \o "C (programming language))**
* **[Cω](https://en.wikipedia.org/wiki/C%CF%89" \o "Cω) – C Omega, a research language extending C#, uses asynchronous communication**
* **[Clojure](https://en.wikipedia.org/wiki/Clojure" \o "Clojure) – a dialect of [Lisp](https://en.wikipedia.org/wiki/Lisp_%28programming_language%29" \o "Lisp (programming language)) for the [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine" \o "Java virtual machine)**
* **[Chapel](https://en.wikipedia.org/wiki/Chapel_%28programming_language%29" \o "Chapel (programming language))**
* **[Co-array Fortran](https://en.wikipedia.org/wiki/Co-array_Fortran" \o "Co-array Fortran)**
* **[Concurrent Pascal](https://en.wikipedia.org/wiki/Concurrent_Pascal" \o "Concurrent Pascal) (by Brinch-Hansen)**
* **[Curry](https://en.wikipedia.org/wiki/Curry_%28programming_language%29" \o "Curry (programming language))**
* **[E](https://en.wikipedia.org/wiki/E_%28programming_language%29" \o "E (programming language)) – uses promises, ensures deadlocks cannot occur**
* **[Eiffel](https://en.wikipedia.org/wiki/Eiffel_%28programming_language%29" \o "Eiffel (programming language)) (through the [SCOOP](https://en.wikipedia.org/wiki/SCOOP_%28software%29" \o "SCOOP (software)) mechanism, Simple Concurrent Object-Oriented Computation)**
* **[Erlang](https://en.wikipedia.org/wiki/Erlang_%28programming_language%29" \o "Erlang (programming language)) – uses asynchronous message passing with nothing shared**
* **[Go](https://en.wikipedia.org/wiki/Go_%28programming_language%29" \o "Go (programming language))**
* **[Java](https://en.wikipedia.org/wiki/Java_%28programming_language%29" \o "Java (programming language))** 
  + **[Join Java](https://en.wikipedia.org/wiki/Join_Java" \o "Join Java) – concurrent language based on Java**
  + **[X10](https://en.wikipedia.org/wiki/X10_%28programming_language%29" \o "X10 (programming language))**
* **[Julia](https://en.wikipedia.org/wiki/Julia_%28programming_language%29" \o "Julia (programming language))**
* **[Join-calculus](https://en.wikipedia.org/wiki/Join-calculus" \o "Join-calculus)**
* **[Joule](https://en.wikipedia.org/wiki/Joule_%28programming_language%29" \o "Joule (programming language)) – dataflow language, communicates by message passing**
* **[Limbo](https://en.wikipedia.org/wiki/Limbo_%28programming_language%29" \o "Limbo (programming language)) – relative of [Alef](https://en.wikipedia.org/wiki/Alef_%28programming_language%29" \o "Alef (programming language)), used for systems programming in [Inferno (operating system)](https://en.wikipedia.org/wiki/Inferno_%28operating_system%29" \o "Inferno (operating system))**
* **[MultiLisp](https://en.wikipedia.org/wiki/MultiLisp" \o "MultiLisp) – [Scheme](https://en.wikipedia.org/wiki/Scheme_%28programming_language%29" \o "Scheme (programming language)) variant extended to support parallelism**
* **[occam](https://en.wikipedia.org/wiki/Occam_%28programming_language%29" \o "Occam (programming language)) – influenced heavily by [Communicating Sequential Processes](https://en.wikipedia.org/wiki/Communicating_sequential_processes" \o "Communicating sequential processes) (CSP)** 
  + **[occam-π](https://en.wikipedia.org/wiki/Occam-%CF%80" \o "Occam-π) – a modern variant of [occam](https://en.wikipedia.org/wiki/Occam_%28programming_language%29" \o "Occam (programming language)), which incorporates ideas from Milner's [π-calculus](https://en.wikipedia.org/wiki/Pi-calculus" \o "Pi-calculus)**
* **[Orc](https://en.wikipedia.org/wiki/Orc_%28programming_language%29" \o "Orc (programming language))**
* **[Oz](https://en.wikipedia.org/wiki/Oz_%28programming_language%29" \o "Oz (programming language)) – multiparadigm language, supports shared-state and message-passing concurrency, and futures, and Mozart Programming System [cross-platform](https://en.wikipedia.org/wiki/Cross-platform" \o "Cross-platform) Oz**
* **[Pict](https://en.wikipedia.org/wiki/Pict_%28programming_language%29" \o "Pict (programming language)) – essentially an executable implementation of Milner's [π-calculus](https://en.wikipedia.org/wiki/Pi-calculus" \o "Pi-calculus)**
* **[Rust](https://en.wikipedia.org/wiki/Rust_%28programming_language%29" \o "Rust (programming language)) – actor-based**
* **[SALSA](https://en.wikipedia.org/wiki/SALSA_%28programming_language%29" \o "SALSA (programming language)) – actor language with token-passing, join, and first-class continuations for distributed computing over the Internet**
* **[Scala](https://en.wikipedia.org/wiki/Scala_%28programming_language%29" \o "Scala (programming language)) – implements Erlang-style [actors](https://en.wikipedia.org/wiki/Actor_model" \o "Actor model) on the JVM**
* **[SequenceL](https://en.wikipedia.org/wiki/SequenceL" \o "SequenceL) – purely functional, automatically parallelizing and race-free**
* **[SR](https://en.wikipedia.org/wiki/SR_%28programming_language%29" \o "SR (programming language)) – research language**
* **[Unified Parallel C](https://en.wikipedia.org/wiki/Unified_Parallel_C" \o "Unified Parallel C)**
* **[XProc](https://en.wikipedia.org/wiki/XProc" \o "XProc) – XML processing language, enabling concurrency**

**4. Curly-bracket languages**

* **[ABCL/c+](https://en.wikipedia.org/wiki/Actor-Based_Concurrent_Language" \o "Actor-Based Concurrent Language)**
* **[AutoHotkey](https://en.wikipedia.org/wiki/AutoHotkey" \o "AutoHotkey)**
* **[AWK](https://en.wikipedia.org/wiki/AWK" \o "AWK)**
* **[B](https://en.wikipedia.org/wiki/B_%28programming_language%29" \o "B (programming language))**
* **[bc](https://en.wikipedia.org/wiki/Bc_%28programming_language%29" \o "Bc (programming language))**
* **[BCPL](https://en.wikipedia.org/wiki/BCPL" \o "BCPL)**
* **[C](https://en.wikipedia.org/wiki/C_%28programming_language%29" \o "C (programming language)) – developed circa 1970 at [Bell Labs](https://en.wikipedia.org/wiki/Bell_Labs" \o "Bell Labs)**
* **[C++](https://en.wikipedia.org/wiki/C%2B%2B" \o "C++)**
* **[C#](https://en.wikipedia.org/wiki/C_Sharp_%28programming_language%29" \o "C Sharp (programming language))**
* **[Ceylon](https://en.wikipedia.org/wiki/Ceylon_%28programming_language%29" \o "Ceylon (programming language))**
* **[ChucK](https://en.wikipedia.org/wiki/ChucK" \o "ChucK) – audio programming language**
* **[Cilk](https://en.wikipedia.org/wiki/Cilk" \o "Cilk) – concurrent C for multithreaded parallel programming**
* **[COFFEE](https://en.wikipedia.org/wiki/COFFEE_%28Cinema_4D%29" \o "COFFEE (Cinema 4D))**
* **[Cyclone](https://en.wikipedia.org/wiki/Cyclone_%28programming_language%29" \o "Cyclone (programming language)) – a safer C variant**
* **[D](https://en.wikipedia.org/wiki/D_%28programming_language%29" \o "D (programming language))**
* **[Dart](https://en.wikipedia.org/wiki/Dart_%28programming_language%29" \o "Dart (programming language))**
* **[DASL](https://en.wikipedia.org/wiki/Distributed_Application_Specification_Language" \o "Distributed Application Specification Language) – based on Java**
* **[E](https://en.wikipedia.org/wiki/E_%28programming_language%29" \o "E (programming language))**
* **[eC](https://en.wikipedia.org/wiki/EC_%28programming_language%29" \o "EC (programming language))**
* **[ECMAScript](https://en.wikipedia.org/wiki/ECMAScript" \o "ECMAScript)** 
  + **[ActionScript](https://en.wikipedia.org/wiki/ActionScript" \o "ActionScript)**
  + **[ECMAScript for XML](https://en.wikipedia.org/wiki/ECMAScript_for_XML" \o "ECMAScript for XML)**
  + **[JavaScript](https://en.wikipedia.org/wiki/JavaScript" \o "JavaScript)**
  + **[JScript](https://en.wikipedia.org/wiki/JScript" \o "JScript)**
  + **[TypeScript](https://en.wikipedia.org/wiki/TypeScript" \o "TypeScript)**
* **GML ([Game Maker Language](https://en.wikipedia.org/wiki/GameMaker:_Studio" \o "GameMaker: Studio))**
* **[GLSL](https://en.wikipedia.org/wiki/GLSL" \o "GLSL)**
* **[ICI](https://en.wikipedia.org/wiki/ICI_%28programming_language%29" \o "ICI (programming language))**
* **[Java](https://en.wikipedia.org/wiki/Java_%28programming_language%29" \o "Java (programming language))** 
  + **[Processing](https://en.wikipedia.org/wiki/Processing_%28programming_language%29" \o "Processing (programming language))**
  + **[Groovy](https://en.wikipedia.org/wiki/Groovy_%28programming_language%29" \o "Groovy (programming language))**
  + **[Join Java](https://en.wikipedia.org/wiki/Join_Java" \o "Join Java)**
  + **[Kotlin](https://en.wikipedia.org/wiki/Kotlin_%28programming_language%29" \o "Kotlin (programming language))**
  + **[Tea](https://en.wikipedia.org/wiki/Tea_%28programming_language%29" \o "Tea (programming language))**
  + **[X10](https://en.wikipedia.org/wiki/X10_%28programming_language%29" \o "X10 (programming language))**
* **[LPC](https://en.wikipedia.org/wiki/LPC_%28programming_language%29" \o "LPC (programming language))**
* **[MSL](https://en.wikipedia.org/wiki/MIRC_scripting_language" \o "MIRC scripting language)**
* **[MEL](https://en.wikipedia.org/wiki/Maya_Embedded_Language" \o "Maya Embedded Language)**
* **[Nemerle](https://en.wikipedia.org/wiki/Nemerle" \o "Nemerle) – combines C# and ML features, provides syntax extension abilities**
* **[PCASTL](https://en.wikipedia.org/wiki/PCASTL" \o "PCASTL)**
* **[Perl](https://en.wikipedia.org/wiki/Perl" \o "Perl)**
* **[PHP](https://en.wikipedia.org/wiki/PHP" \o "PHP)**
* **[Pico](https://en.wikipedia.org/wiki/Pico_%28programming_language%29" \o "Pico (programming language))**
* **[Pike](https://en.wikipedia.org/wiki/Pike_%28programming_language%29" \o "Pike (programming language))**
* **[R](https://en.wikipedia.org/wiki/R_%28programming_language%29" \o "R (programming language))**
* **[Rust](https://en.wikipedia.org/wiki/Rust_%28programming_language%29" \o "Rust (programming language))**
* **[S-Lang](https://en.wikipedia.org/wiki/S-Lang_%28programming_library%29" \o "S-Lang (programming library))**
* **[Scala](https://en.wikipedia.org/wiki/Scala_%28programming_language%29" \o "Scala (programming language))**
* **[sed](https://en.wikipedia.org/wiki/Sed" \o "Sed)**
* **[SuperCollider](https://en.wikipedia.org/wiki/SuperCollider" \o "SuperCollider)**
* **[Swift](https://en.wikipedia.org/wiki/Swift_%28programming_language%29" \o "Swift (programming language))**
* **[UnrealScript](https://en.wikipedia.org/wiki/UnrealScript" \o "UnrealScript)**
* **[Windows PowerShell](https://en.wikipedia.org/wiki/Windows_PowerShell" \o "Windows PowerShell) ([Microsoft](https://en.wikipedia.org/wiki/Microsoft" \o "Microsoft) [.NET](https://en.wikipedia.org/wiki/.NET_Framework" \o ".NET Framework)-based CLI)**

**5. DATA FLOW LANGUAGES**

* **[Hartmann pipelines](https://en.wikipedia.org/wiki/Hartmann_pipeline" \o "Hartmann pipeline)**
* **G (used in [LabVIEW](https://en.wikipedia.org/wiki/LabVIEW" \o "LabVIEW))**
* **[Lucid](https://en.wikipedia.org/wiki/Lucid_%28programming_language%29" \o "Lucid (programming language))**
* **[Max](https://en.wikipedia.org/wiki/Max_%28software%29" \o "Max (software))**
* **[Oz](https://en.wikipedia.org/wiki/Oz_%28programming_language%29" \o "Oz (programming language))**
* **[Prograph](https://en.wikipedia.org/wiki/Prograph" \o "Prograph)**
* **[Pure Data](https://en.wikipedia.org/wiki/Pure_Data" \o "Pure Data)**
* **[Reaktor](https://en.wikipedia.org/wiki/Reaktor" \o "Reaktor)**
* **[StreamBase StreamSQL EventFlow](https://en.wikipedia.org/wiki/StreamBase_Systems" \l "StreamSQL_EventFlow_Language" \o "StreamBase Systems)**
* **[VEE](https://en.wikipedia.org/wiki/Agilent_VEE" \o "Agilent VEE)**
* **[VHDL](https://en.wikipedia.org/wiki/VHDL" \o "VHDL)**
* **[VisSim](https://en.wikipedia.org/wiki/VisSim" \o "VisSim)**
* **[WebMethods Flow](https://en.wikipedia.org/wiki/WebMethods_Flow" \o "WebMethods Flow)**

1. **Embeddable languages**

**Server side**

* **[PHP](https://en.wikipedia.org/wiki/PHP" \o "PHP)**
* **[VBScript](https://en.wikipedia.org/wiki/VBScript" \o "VBScript)**
* **[SMX](https://en.wikipedia.org/wiki/SMX_%28computer_language%29" \o "SMX (computer language)) – dedicated to web pages**
* **[WebDNA](https://en.wikipedia.org/wiki/WebDNA" \o "WebDNA) – dedicated to database-driven websites**

**Client side**

* **[ActionScript](https://en.wikipedia.org/wiki/ActionScript" \o "ActionScript)**
* **[Java](https://en.wikipedia.org/wiki/Java_%28programming_language%29" \o "Java (programming language))**
* **[JavaScript](https://en.wikipedia.org/wiki/JavaScript" \o "JavaScript)**
* **[ECMAScript](https://en.wikipedia.org/wiki/ECMAScript" \o "ECMAScript)**
* **[JScript](https://en.wikipedia.org/wiki/JScript" \o "JScript)**
* **[VBScript](https://en.wikipedia.org/wiki/VBScript" \o "VBScript) (Windows only)**

**In object code**

* **[AngelScript](https://en.wikipedia.org/wiki/AngelScript" \o "AngelScript)**
* **[Ch](https://en.wikipedia.org/wiki/Ch_%28computer_programming%29" \o "Ch (computer programming))**
* **[EEL](https://en.wikipedia.org/wiki/Extensible_Embeddable_Language" \o "Extensible Embeddable Language)**
* **[Io](https://en.wikipedia.org/wiki/Io_%28programming_language%29" \o "Io (programming language))**
* **[Julia](https://en.wikipedia.org/wiki/Julia_%28programming_language%29" \o "Julia (programming language))**
* **[Lua](https://en.wikipedia.org/wiki/Lua_%28programming_language%29" \o "Lua (programming language))**
* **[MiniD](https://en.wikipedia.org/wiki/MiniD" \o "MiniD)**
* **[Python](https://en.wikipedia.org/wiki/Python_%28programming_language%29" \o "Python (programming language))**
* **[Ruby](https://en.wikipedia.org/wiki/Ruby_%28programming_language%29" \o "Ruby (programming language)) (via [mruby](https://en.wikipedia.org/wiki/Mruby" \o "Mruby))**
* **[Squirrel](https://en.wikipedia.org/wiki/Squirrel_%28programming_language%29" \o "Squirrel (programming language))**
* **[Tcl](https://en.wikipedia.org/wiki/Tcl" \o "Tcl)**

1. **Educational languages**

* **[Alice](https://en.wikipedia.org/wiki/Alice_%28software%29" \o "Alice (software))**
* **[Blockly](https://en.wikipedia.org/wiki/Blockly" \o "Blockly)**
* **[Blue](https://en.wikipedia.org/wiki/Blue_%28programming_language%29" \o "Blue (programming language))**
* **[COMAL](https://en.wikipedia.org/wiki/COMAL_%28programming_language%29" \o "COMAL (programming language))**
* **[Elan](https://en.wikipedia.org/wiki/ELAN_%28programming_language%29" \o "ELAN (programming language))**
* **[Logo](https://en.wikipedia.org/wiki/Logo_%28programming_language%29" \o "Logo (programming language))**
* **[KTurtle](https://en.wikipedia.org/wiki/KTurtle" \o "KTurtle)**
* **[Modula-2](https://en.wikipedia.org/wiki/Modula-2" \o "Modula-2)**
* **[Pascal](https://en.wikipedia.org/wiki/Pascal_%28programming_language%29" \o "Pascal (programming language))**

1. **Esoteric languages**

* **[Beatnik](https://en.wikipedia.org/wiki/Beatnik_%28programming_language%29" \o "Beatnik (programming language))**
* **[Befunge](https://en.wikipedia.org/wiki/Befunge" \o "Befunge)**
* **[Brainfuck](https://en.wikipedia.org/wiki/Brainfuck" \o "Brainfuck)**
* **[Chef](https://en.wikipedia.org/wiki/Chef_%28programming_language%29" \o "Chef (programming language))**
* **[INTERCAL](https://en.wikipedia.org/wiki/INTERCAL" \o "INTERCAL)**
* **[LOLCODE](https://en.wikipedia.org/wiki/LOLCODE" \o "LOLCODE)**
* **[Malbolge](https://en.wikipedia.org/wiki/Malbolge" \o "Malbolge)**

1. **Extension languages**

* **[Ateji PX](https://en.wikipedia.org/wiki/Ateji_PX" \o "Ateji PX) – an extension of the Java language for parallelism**
* **[AutoLISP](https://en.wikipedia.org/wiki/AutoLISP" \o "AutoLISP) (specific to [AutoCAD](https://en.wikipedia.org/wiki/AutoCAD" \o "AutoCAD))**
* **[BeanShell](https://en.wikipedia.org/wiki/BeanShell" \o "BeanShell)**
* **[CAL](https://en.wikipedia.org/wiki/Cakewalk_%28sequencer%29" \l "Features" \o "Cakewalk (sequencer))**
* **[C/AL(C/SIDE)](https://en.wikipedia.org/wiki/C/AL" \o "C/AL)**
* **[Guile](https://en.wikipedia.org/wiki/GNU_Guile" \o "GNU Guile)**
* **[Emacs Lisp](https://en.wikipedia.org/wiki/Emacs_Lisp" \o "Emacs Lisp)**
* **[JavaScript](https://en.wikipedia.org/wiki/JavaScript" \o "JavaScript) and some dialects (e.g. [JScript](https://en.wikipedia.org/wiki/JScript" \o "JScript))**
* **[Lua](https://en.wikipedia.org/wiki/Lua_%28programming_language%29" \o "Lua (programming language)) – e.g. embedded in many games**
* **[OpenCL](https://en.wikipedia.org/wiki/OpenCL" \o "OpenCL) – an extension of C and C++ to use the GPU and parallel extensions of the CPU**

**10. Fourth-generation languages**

* **[ABAP](https://en.wikipedia.org/wiki/ABAP" \o "ABAP)**
* **[CorVision](https://en.wikipedia.org/wiki/CorVision" \o "CorVision)**
* **[CSC](https://en.wikipedia.org/wiki/Computer_Sciences_Corporation" \o "Computer Sciences Corporation)'s GraphTalk**
* **[Easytrieve](https://en.wikipedia.org/wiki/Easytrieve" \o "Easytrieve) report generator (now CA-Easytrieve Plus)**
* **[FOCUS](https://en.wikipedia.org/wiki/FOCUS" \o "FOCUS)**
* **[IBM Informix-4GL](https://en.wikipedia.org/wiki/IBM_Informix-4GL" \o "IBM Informix-4GL) / [Aubit-4GL](https://en.wikipedia.org/wiki/Aubit-4GL" \o "Aubit-4GL)**
* **[LANSA](https://en.wikipedia.org/wiki/LANSA_%28development_environment%29" \o "LANSA (development environment))**
* **[LINC 4GL](https://en.wikipedia.org/wiki/LINC_4GL" \o "LINC 4GL)**
* **[MAPPER](https://en.wikipedia.org/wiki/MAPPER" \o "MAPPER) ([Unisys/Sperry](https://en.wikipedia.org/wiki/Unisys" \o "Unisys)) – now part of BIS**
* **[MARK-IV](https://en.wikipedia.org/wiki/MARK_IV_%28software%29" \o "MARK IV (software)) ([Sterling/Informatics](https://en.wikipedia.org/wiki/Sterling_Software" \o "Sterling Software)) now VISION:BUILDER of CA**

**11.Functional languages**

**See also: [Category:Functional languages](https://en.wikipedia.org/wiki/Category:Functional_languages" \o "Category:Functional languages)**

**[Functional programming](https://en.wikipedia.org/wiki/Functional_programming" \o "Functional programming) languages define programs and subroutines as mathematical functions. Many so-called functional languages are "impure", containing imperative features. Many functional languages are tied to mathematical calculation tools. Functional languages include:**

**Pure**

* **[Agda](https://en.wikipedia.org/wiki/Agda_%28programming_language%29" \o "Agda (programming language))**
* **[Charity](https://en.wikipedia.org/wiki/Charity_%28programming_language%29" \o "Charity (programming language))**
* **[Clean](https://en.wikipedia.org/wiki/Clean_%28programming_language%29" \o "Clean (programming language))**
* **[Coq (Gallina)](https://en.wikipedia.org/wiki/Coq" \o "Coq)**
* **[Curry](https://en.wikipedia.org/wiki/Curry_%28programming_language%29" \o "Curry (programming language))**
* **[Elm](https://en.wikipedia.org/wiki/Elm_%28programming_language%29" \o "Elm (programming language))**

**Impure**

* **[APL](https://en.wikipedia.org/wiki/APL_%28programming_language%29" \o "APL (programming language))**
* **[ATS](https://en.wikipedia.org/wiki/ATS_%28programming_language%29" \o "ATS (programming language))**
* **[CAL](https://en.wikipedia.org/wiki/CAL_%28Joss_family%29" \o "CAL (Joss family))**
* **[C](https://en.wikipedia.org/wiki/C_%28programming_language%29" \o "C (programming language))**
* **[C++](https://en.wikipedia.org/wiki/C%2B%2B" \o "C++) (since [C++11](https://en.wikipedia.org/wiki/C%2B%2B11" \o "C++11))**
* **[C#](https://en.wikipedia.org/wiki/C_Sharp_%28programming_language%29" \o "C Sharp (programming language))**

**12. Hardware description languages**

**HDLs for analog circuit design**

* **[Verilog-AMS](https://en.wikipedia.org/wiki/Verilog-AMS" \o "Verilog-AMS) (Verilog for Analog and Mixed-Signal)**
* **[VHDL-AMS](https://en.wikipedia.org/wiki/VHDL-AMS" \o "VHDL-AMS) (VHDL with Analog/Mixed-Signal extension)**

**HDLs for digital circuit design**

* **[Advanced Boolean Expression Language](https://en.wikipedia.org/wiki/Advanced_Boolean_Expression_Language" \o "Advanced Boolean Expression Language) (ABEL)**
* **[Altera Hardware Description Language](https://en.wikipedia.org/wiki/Altera_Hardware_Description_Language" \o "Altera Hardware Description Language) (AHDL)**
* **[Bluespec](https://en.wikipedia.org/wiki/Bluespec" \o "Bluespec)**
* **[Chisel](https://chisel.eecs.berkeley.edu)**
* **[Confluence](https://en.wikipedia.org/wiki/Confluence" \o "Confluence)**
* **[ELLA](https://en.wikipedia.org/wiki/ELLA_%28programming_language%29" \o "ELLA (programming language))**
* **[ESys.net](https://en.wikipedia.org/wiki/ESys.net" \o "ESys.net)**
* **[Handel-C](https://en.wikipedia.org/wiki/Handel-C" \o "Handel-C)**
* **[HHDL](https://en.wikipedia.org/wiki/HHDL" \o "HHDL)**

**14. Imperative languages**

**Imperative programming languages may be multi-paradigm and appear in other classifications. Here is a list of programming languages that follow the [imperative paradigm](https://en.wikipedia.org/wiki/Imperative_paradigm" \o "Imperative paradigm):**

* **[Ada](https://en.wikipedia.org/wiki/Ada_%28programming_language%29" \o "Ada (programming language))**
* **[ALGOL](https://en.wikipedia.org/wiki/ALGOL" \o "ALGOL)**
* **[BASIC](https://en.wikipedia.org/wiki/BASIC" \o "BASIC)**
* **[Blue](https://en.wikipedia.org/wiki/Blue_%28programming_language%29" \o "Blue (programming language))**
* **[C](https://en.wikipedia.org/wiki/C_%28programming_language%29" \o "C (programming language))**
* **[C++](https://en.wikipedia.org/wiki/C%2B%2B" \o "C++)**
* **[C#](https://en.wikipedia.org/wiki/C_Sharp_%28programming_language%29" \o "C Sharp (programming language))**
* **[Ceylon](https://en.wikipedia.org/wiki/Ceylon_%28programming_language%29" \o "Ceylon (programming language))**
* **[COBOL](https://en.wikipedia.org/wiki/COBOL" \o "COBOL)**

**15. Interactive mode languages**

* **[APL](https://en.wikipedia.org/wiki/APL_%28programming_language%29" \o "APL (programming language))**
* **[BASIC](https://en.wikipedia.org/wiki/BASIC" \o "BASIC) (some dialects)**
* **[Clojure](https://en.wikipedia.org/wiki/Clojure" \o "Clojure)**
* **[Common Lisp](https://en.wikipedia.org/wiki/Common_Lisp" \o "Common Lisp)**
* **[Dart](https://en.wikipedia.org/wiki/Dart_%28programming_language%29" \o "Dart (programming language)) (with Observatory or Dartium's developer tools)**
* **[Erlang](https://en.wikipedia.org/wiki/Erlang_%28programming_language%29" \o "Erlang (programming language))**
* **[F#](https://en.wikipedia.org/wiki/F_Sharp_%28programming_language%29" \o "F Sharp (programming language))**
* **[Forth](https://en.wikipedia.org/wiki/Forth_%28programming_language%29" \o "Forth (programming language))**
* **[FPr](https://en.wikipedia.org/wiki/FPr_%28programming_language%29" \o "FPr (programming language))**
* **[Fril](https://en.wikipedia.org/wiki/Fril" \o "Fril)**
* **[GAUSS](https://en.wikipedia.org/wiki/GAUSS_%28software%29" \o "GAUSS (software))**
* **[Groovy](https://en.wikipedia.org/wiki/Groovy_%28programming_language%29" \o "Groovy (programming language))**

**17. Interpreted languages**

**[Interpreted languages](https://en.wikipedia.org/wiki/Interpreted_language" \o "Interpreted language) are programming languages in which programs may be executed from source code form, by an interpreter. Theoretically, any language can be compiled or interpreted, so the term \*interpreted language\* generally refers to languages that are commonly interpreted rather than compiled.**

* **[Ant](https://en.wikipedia.org/wiki/Apache_Ant" \o "Apache Ant)**
* **[APL](https://en.wikipedia.org/wiki/APL_%28programming_language%29" \o "APL (programming language))**
* **[AutoHotkey](https://en.wikipedia.org/wiki/AutoHotkey" \o "AutoHotkey) scripting language**
* **[AutoIt](https://en.wikipedia.org/wiki/AutoIt" \o "AutoIt) scripting language**
* **[BASIC](https://en.wikipedia.org/wiki/BASIC" \o "BASIC) (some dialects)**
* **[DATABUS](https://en.wikipedia.org/wiki/DATABUS_%28programming_language%29" \o "DATABUS (programming language)) (later versions added optional compiling)**
* **[DM](https://en.wikipedia.org/wiki/DM_%28computing%29" \o "DM (computing))**
* **[Eiffel](https://en.wikipedia.org/wiki/Eiffel_%28programming_language%29" \o "Eiffel (programming language)) (via "Melting Ice Technology" in [EiffelStudio](https://en.wikipedia.org/wiki/EiffelStudio" \o "EiffelStudio))**
* **[Mathematica](https://en.wikipedia.org/wiki/Mathematica" \o "Mathematica)**
* **[MATLAB](https://en.wikipedia.org/wiki/MATLAB" \o "MATLAB)**
* **[Oriel](https://en.wikipedia.org/wiki/Oriel_%28scripting_language%29" \o "Oriel (scripting language))**
* **[Pascal](https://en.wikipedia.org/wiki/Pascal_%28programming_language%29" \o "Pascal (programming language)) (early implementations)**
* **[PCASTL](https://en.wikipedia.org/wiki/PCASTL" \o "PCASTL)**

**18. Iterative languages**

**Iterative languages are built around or offering [generators](https://en.wikipedia.org/wiki/Generator_%28computer_science%29" \o "Generator (computer science)).**

* **[Aldor](https://en.wikipedia.org/wiki/Aldor" \o "Aldor)**
* **[Alphard](https://en.wikipedia.org/wiki/Alphard_%28programming_language%29" \o "Alphard (programming language))**
* **[C#](https://en.wikipedia.org/wiki/Generator_%28computer_science%29" \l "C.23" \o "Generator (computer science))**
* **[CLU](https://en.wikipedia.org/wiki/CLU_%28programming_language%29" \o "CLU (programming language))**
* **[Cobra](https://en.wikipedia.org/wiki/Cobra_%28programming_language%29" \o "Cobra (programming language))**
* **[ECMAScript](https://en.wikipedia.org/wiki/ECMAScript" \o "ECMAScript), as of version 6**
* **[Eiffel](https://en.wikipedia.org/wiki/Eiffel_%28programming_language%29" \o "Eiffel (programming language)), through "agents"**
* **[Icon](https://en.wikipedia.org/wiki/Icon_%28programming_language%29" \o "Icon (programming language))**
* **[IPL-v](https://en.wikipedia.org/wiki/Information_Processing_Language" \o "Information Processing Language)**
* **[Julia](https://en.wikipedia.org/wiki/Julia_%28programming_language%29" \o "Julia (programming language))**

**19. little languages**

* **[awk](https://en.wikipedia.org/wiki/AWK" \o "AWK) – can serve as a prototyping language for [C](https://en.wikipedia.org/wiki/C_%28programming_language%29" \o "C (programming language)) (shares similar syntax)**
* **[Comet](https://en.wikipedia.org/wiki/Comet_%28programming_language%29" \o "Comet (programming language)) – used to solve complex combinatorial [optimization](https://en.wikipedia.org/wiki/Program_optimization" \o "Program optimization) problems in areas such as [resource allocation](https://en.wikipedia.org/wiki/Resource_allocation" \o "Resource allocation) and [scheduling](https://en.wikipedia.org/wiki/Scheduling_%28computing%29" \o "Scheduling (computing))**
* **[SQL](https://en.wikipedia.org/wiki/SQL" \o "SQL) – has only a few keywords, and not all the constructs needed for a full programming language[[1]](https://en.wikipedia.org/wiki/List_of_programming_languages_by_type" \l "cite_note-1) – many database management systems extend SQL with additional constructs as a [stored procedure](https://en.wikipedia.org/wiki/Stored_procedure" \o "Stored procedure) language**

**20. Logic-based languages**

* **[ALF](https://en.wikipedia.org/wiki/Algebraic_Logic_Functional_%28programming_language%29" \o "Algebraic Logic Functional (programming language))**
* **[Alma-0](https://en.wikipedia.org/wiki/Alma-0" \o "Alma-0)**
* **[CLACL (CLAC-Language)](https://en.wikipedia.org/wiki/CLACL_%28programming_language%29" \o "CLACL (programming language))**
* **[Curry](https://en.wikipedia.org/wiki/Curry_%28programming_language%29" \o "Curry (programming language))**
* **[Fril](https://en.wikipedia.org/wiki/Fril" \o "Fril)**
* **[Janus](https://en.wikipedia.org/wiki/Janus_%28concurrent_constraint_programming_language%29" \o "Janus (concurrent constraint programming language))**

**21. Machine languages**

* **[IBM System/360](https://en.wikipedia.org/wiki/IBM_System/360" \o "IBM System/360) and successors, including [z/Architecture](https://en.wikipedia.org/wiki/Z/Architecture" \o "Z/Architecture)**
* **[MIPS](https://en.wikipedia.org/wiki/MIPS_instruction_set" \o "MIPS instruction set)**
* **[Motorola 6800](https://en.wikipedia.org/wiki/Motorola_6800" \o "Motorola 6800)**
* **[Motorola 68000 family](https://en.wikipedia.org/wiki/Motorola_68000_family" \o "Motorola 68000 family) (CPUs used in early [Apple Macintosh](https://en.wikipedia.org/wiki/Apple_Macintosh" \o "Apple Macintosh) and early [Sun](https://en.wikipedia.org/wiki/Sun_Microsystems" \o "Sun Microsystems) computers)**
* **[MOS Technology](https://en.wikipedia.org/wiki/MOS_Technology" \o "MOS Technology) [65xx](https://en.wikipedia.org/wiki/MOS_Technology_65xx" \o "MOS Technology 65xx)** 
  + **[6502](https://en.wikipedia.org/wiki/MOS_Technology_6502" \o "MOS Technology 6502) (CPU for [VIC-20](https://en.wikipedia.org/wiki/Commodore_VIC-20" \o "Commodore VIC-20), [Apple II](https://en.wikipedia.org/wiki/Apple_II_family" \o "Apple II family), and [Atari 800](https://en.wikipedia.org/wiki/Atari_8-bit_family" \o "Atari 8-bit family))**
  + **[6510](https://en.wikipedia.org/wiki/MOS_Technology_6510" \o "MOS Technology 6510) (CPU for [Commodore 64](https://en.wikipedia.org/wiki/Commodore_64" \o "Commodore 64))**
  + **[Western Design Center](https://en.wikipedia.org/wiki/Western_Design_Center" \o "Western Design Center) [65816/65802](https://en.wikipedia.org/wiki/WDC_65816/65802" \o "WDC 65816/65802) (CPU for [Apple IIGS](https://en.wikipedia.org/wiki/Apple_IIGS" \o "Apple IIGS) and (variant) [Super Nintendo Entertainment System](https://en.wikipedia.org/wiki/Super_Nintendo_Entertainment_System" \o "Super Nintendo Entertainment System))**
* **National [NS320xx](https://en.wikipedia.org/wiki/NS320xx" \o "NS320xx)**
* **[Power Architecture](https://en.wikipedia.org/wiki/Power_Architecture" \o "Power Architecture)** 
  + **[POWER](https://en.wikipedia.org/wiki/IBM_POWER_Instruction_Set_Architecture" \o "IBM POWER Instruction Set Architecture), first used in the [IBM RS/6000](https://en.wikipedia.org/wiki/IBM_RS/6000" \o "IBM RS/6000)**
  + **[PowerPC](https://en.wikipedia.org/wiki/PowerPC" \o "PowerPC) – used in [Power Macintosh](https://en.wikipedia.org/wiki/Power_Macintosh" \o "Power Macintosh) and the technology is used in many older generation [game consoles](https://en.wikipedia.org/wiki/PowerPC" \l "Gaming_consoles" \o "PowerPC)**
* **Sun, [Oracle](https://en.wikipedia.org/wiki/Oracle_Corporation" \o "Oracle Corporation) [SPARC](https://en.wikipedia.org/wiki/SPARC" \o "SPARC)**
* **[MCST](https://en.wikipedia.org/wiki/Moscow_Center_of_SPARC_Technologies" \o "Moscow Center of SPARC Technologies) [Elbrus 2000](https://en.wikipedia.org/wiki/Elbrus_2000" \o "Elbrus 2000)**

**22. Macro languages**

**Textual substitution macro languages**

* **[cpp](https://en.wikipedia.org/wiki/C_preprocessor" \o "C preprocessor) (the C preprocessor)**
* **[m4](https://en.wikipedia.org/wiki/M4_%28computer_language%29" \o "M4 (computer language)) (originally from AT&T, bundled with Unix)**

**23. Metaprogramming languages**

* **[C++](https://en.wikipedia.org/wiki/C%2B%2B" \o "C++)**
* **[Curl](https://en.wikipedia.org/wiki/Curl_%28programming_language%29" \o "Curl (programming language))**
* **[D](https://en.wikipedia.org/wiki/D_%28programming_language%29" \o "D (programming language))**
* **[eC](https://en.wikipedia.org/wiki/EC_%28programming_language%29" \o "EC (programming language))**
* **[Elixir](https://en.wikipedia.org/wiki/Elixir_%28programming_language%29" \o "Elixir (programming language))**
* **[Forth](https://en.wikipedia.org/wiki/Forth_%28programming_language%29" \o "Forth (programming language))**
* **[Groovy](https://en.wikipedia.org/wiki/Groovy_%28programming_language%29" \o "Groovy (programming language))**
* **[Haskell](https://en.wikipedia.org/wiki/Haskell_%28programming_language%29" \o "Haskell (programming language))**
* **[Julia](https://en.wikipedia.org/wiki/Julia_%28programming_language%29" \o "Julia (programming language))**
* **[Lisp](https://en.wikipedia.org/wiki/Lisp_%28programming_language%29" \o "Lisp (programming language))**
* **[Lua](https://en.wikipedia.org/wiki/Lua_%28programming_language%29" \o "Lua (programming language))**
* **[Maude system](https://en.wikipedia.org/wiki/Maude_system" \o "Maude system)**

**24. Multiparadigm languages**

* **[Ada](https://en.wikipedia.org/wiki/Ada_%28programming_language%29" \o "Ada (programming language)) ([concurrent](https://en.wikipedia.org/wiki/Parallel_computing" \o "Parallel computing), [distributed](https://en.wikipedia.org/wiki/Distributed_computing" \o "Distributed computing), [generic](https://en.wikipedia.org/wiki/Generic_programming" \o "Generic programming) ([template metaprogramming](https://en.wikipedia.org/wiki/Template_metaprogramming" \o "Template metaprogramming)), [imperative](https://en.wikipedia.org/wiki/Imperative_programming" \o "Imperative programming), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming" \o "Object-oriented programming) ([class-based](https://en.wikipedia.org/wiki/Class_%28computer_science%29" \o "Class (computer science))))**
* **[ALF](https://en.wikipedia.org/wiki/Algebraic_Logic_Functional_%28programming_language%29" \o "Algebraic Logic Functional (programming language)) ([functional](https://en.wikipedia.org/wiki/Functional_programming" \o "Functional programming), [logic](https://en.wikipedia.org/wiki/Logic_programming" \o "Logic programming))**
* **[Alma-0](https://en.wikipedia.org/wiki/Alma-0" \o "Alma-0) (constraint, imperative, logic)**
* **[APL](https://en.wikipedia.org/wiki/APL_%28programming_language%29" \o "APL (programming language)) (functional, imperative)**
* **[BETA](https://en.wikipedia.org/wiki/BETA_%28programming_language%29" \o "BETA (programming language)) (functional, imperative, object-oriented (class-based))**
* **[C++](https://en.wikipedia.org/wiki/C%2B%2B" \o "C++) (generic, imperative, object-oriented (class-based), functional)**
* **[C#](https://en.wikipedia.org/wiki/C_Sharp_%28programming_language%29" \o "C Sharp (programming language)) (generic, imperative, object-oriented (class-based), functional, declarative)**
* **[Ceylon](https://en.wikipedia.org/wiki/Ceylon_%28programming_language%29" \o "Ceylon (programming language)) (generic, imperative, object-oriented (class-based), functional, declarative)**
* **[ChucK](https://en.wikipedia.org/wiki/ChucK" \o "ChucK) (imperative, object-oriented, time-based, concurrent, on-the-fly)**
* **[Cobra](https://en.wikipedia.org/wiki/Cobra_%28programming_language%29" \o "Cobra (programming language)) (generic, imperative, object-oriented (class-based), functional, contractual)**
* **[Common Lisp](https://en.wikipedia.org/wiki/Common_Lisp" \o "Common Lisp) (functional, imperative, object-oriented (class-based), [aspect-oriented](https://en.wikipedia.org/wiki/Aspect-oriented_programming" \o "Aspect-oriented programming) (user may add further paradigms, e.g., logic))**
* **[Curl](https://en.wikipedia.org/wiki/Curl_%28programming_language%29" \o "Curl (programming language)) (functional, imperative, object-oriented (class-based), metaprogramming)**
* **[Curry](https://en.wikipedia.org/wiki/Curry_%28programming_language%29" \o "Curry (programming language)) (concurrent, functional, logic)**
* **[F#](https://en.wikipedia.org/wiki/F_Sharp_%28programming_language%29" \o "F Sharp (programming language)) (functional, generic, object-oriented (class-based), language-oriented)**
* **[Fantom](https://en.wikipedia.org/wiki/Fantom_%28programming_language%29" \o "Fantom (programming language)) (functional, object-oriented (class-based))**
* **[FPr](https://en.wikipedia.org/wiki/FPr_%28programming_language%29" \o "FPr (programming language)) (function-level, object-oriented (class-based))**
* **[Go](https://en.wikipedia.org/wiki/Go_%28programming_language%29" \o "Go (programming language)) (functional, object-oriented (class-based), imperative, procedural),**
* **[Groovy](https://en.wikipedia.org/wiki/Groovy_%28programming_language%29" \o "Groovy (programming language)) (functional, object-oriented (class-based),imperative,procedural)**
* **[Harbour](https://en.wikipedia.org/wiki/Harbour_%28software%29" \o "Harbour (software))**
* **[Hop](https://en.wikipedia.org/wiki/Hop_%28software%29" \o "Hop (software))**
* **[J](https://en.wikipedia.org/wiki/J_%28programming_language%29" \o "J (programming language)) (functional, imperative, object-oriented (class-based))**
* **[Julia](https://en.wikipedia.org/wiki/Julia_%28programming_language%29" \o "Julia (programming language)) (imperative, [multiple dispatch](https://en.wikipedia.org/wiki/Multiple_dispatch" \o "Multiple dispatch) ("object-oriented"), functional, metaprogramming)**
* **[LabVIEW](https://en.wikipedia.org/wiki/LabVIEW" \o "LabVIEW) ([dataflow](https://en.wikipedia.org/wiki/Dataflow_programming" \o "Dataflow programming), [visual](https://en.wikipedia.org/wiki/Visual_programming_language" \o "Visual programming language))**
* **[Lava](https://en.wikipedia.org/wiki/Lava_%28programming_language%29" \o "Lava (programming language)) (object-oriented (class-based), visual)**
* **[Leda](https://en.wikipedia.org/wiki/Leda_%28programming_language%29" \o "Leda (programming language)) (functional, imperative, logic, object-oriented (class-based))**
* **[Lua](https://en.wikipedia.org/wiki/Lua_%28programming_language%29" \o "Lua (programming language)) (functional, imperative, object-oriented ([prototype-based](https://en.wikipedia.org/wiki/Prototype-based_programming" \o "Prototype-based programming)))**
* **[Mercury](https://en.wikipedia.org/wiki/Mercury_%28programming_language%29" \o "Mercury (programming language)) (functional, logical, object-oriented)**
* **[Metaobject protocols](https://en.wikipedia.org/wiki/Metaobject" \o "Metaobject) (object-oriented (class-based, prototype-based))**
* **[Nemerle](https://en.wikipedia.org/wiki/Nemerle" \o "Nemerle) (functional, object-oriented (class-based), imperative, metaprogramming)**
* **[Objective-C](https://en.wikipedia.org/wiki/Objective-C" \o "Objective-C) (imperative, object-oriented (class-based), reflective)**
* **[OCaml](https://en.wikipedia.org/wiki/OCaml" \o "OCaml) (functional, imperative, object-oriented (class-based))**
* **[Oz](https://en.wikipedia.org/wiki/Oz_%28programming_language%29" \o "Oz (programming language)) (functional (evaluation: [eager](https://en.wikipedia.org/wiki/Eager_evaluation" \o "Eager evaluation), [lazy](https://en.wikipedia.org/wiki/Lazy_evaluation" \o "Lazy evaluation)), logic, [constraint](https://en.wikipedia.org/wiki/Constraint_programming" \o "Constraint programming), imperative, object-oriented (class-based), concurrent, distributed), and Mozart Programming System [cross-platform](https://en.wikipedia.org/wiki/Cross-platform" \o "Cross-platform) Oz**

**25. Numerical analysis**

* **[AIMMS](https://en.wikipedia.org/wiki/AIMMS" \o "AIMMS)**
* **[AMPL](https://en.wikipedia.org/wiki/AMPL_%28programming_language%29" \o "AMPL (programming language))**
* **[Analytica](https://en.wikipedia.org/wiki/Analytica_%28software%29" \o "Analytica (software))**
* **[GAUSS](https://en.wikipedia.org/wiki/GAUSS_%28software%29" \o "GAUSS (software))**
* **[GAMS](https://en.wikipedia.org/wiki/General_Algebraic_Modeling_System" \o "General Algebraic Modeling System)**
* **[Julia](https://en.wikipedia.org/wiki/Julia_%28programming_language%29" \o "Julia (programming language))**
* **[Klerer-May System](https://en.wikipedia.org/wiki/Klerer-May_System" \o "Klerer-May System)**
* **[Mathematica](https://en.wikipedia.org/wiki/Mathematica" \o "Mathematica)**
* **[MATLAB](https://en.wikipedia.org/wiki/MATLAB" \o "MATLAB)**
* **[PROSE](https://en.wikipedia.org/wiki/PROSE_modeling_language" \o "PROSE modeling language)**
* **[Seneca](https://en.wikipedia.org/wiki/Oberon_%28programming_language%29" \o "Oberon (programming language)) – an [Oberon](https://en.wikipedia.org/wiki/Oberon_%28programming_language%29" \o "Oberon (programming language)) variant**
* **[Wolfram Language](https://en.wikipedia.org/wiki/Wolfram_Language" \o "Wolfram Language)**

**26. Non-English-based languages**

* **[ARLOGO](https://en.wikipedia.org/wiki/ARLOGO" \o "ARLOGO) – [Arabic](https://en.wikipedia.org/wiki/Arabic_language" \o "Arabic language)**
* **[Chinese BASIC](https://en.wikipedia.org/wiki/Chinese_BASIC" \o "Chinese BASIC) – [Chinese](https://en.wikipedia.org/wiki/Chinese_language" \o "Chinese language)**
* **[Fjölnir](https://en.wikipedia.org/wiki/Fj%C3%B6lnir_%28programming_language%29" \o "Fjölnir (programming language)) – [Icelandic](https://en.wikipedia.org/wiki/Icelandic_language" \o "Icelandic language)**
* **[Language Symbolique d'Enseignement](https://en.wikipedia.org/wiki/LSE_%28programming_language%29" \o "LSE (programming language)) – [French](https://en.wikipedia.org/wiki/French_language" \o "French language)**
* **[Lexico](https://en.wikipedia.org/wiki/Lexico_programming_language" \o "Lexico programming language) – [Spanish](https://en.wikipedia.org/wiki/Spanish_language" \o "Spanish language)**
* **[Rapira](https://en.wikipedia.org/wiki/Rapira" \o "Rapira) – [Russian](https://en.wikipedia.org/wiki/Russian_language" \o "Russian language)**

**27. Object-oriented class-based languages**

**[Multiple dispatch](https://en.wikipedia.org/wiki/Multiple_dispatch" \o "Multiple dispatch)**

* **[Common Lisp](https://en.wikipedia.org/wiki/Common_Lisp" \o "Common Lisp)**
* **[Cecil](https://en.wikipedia.org/wiki/Cecil_%28programming_language%29" \o "Cecil (programming language))**
* **[Dylan](https://en.wikipedia.org/wiki/Dylan_%28programming_language%29" \o "Dylan (programming language))**
* **[Julia](https://en.wikipedia.org/wiki/Julia_%28programming_language%29" \o "Julia (programming language))**

**Single dispatch**

* **[ActionScript 3.0](https://en.wikipedia.org/wiki/ActionScript" \o "ActionScript)**
* **[Actor](https://en.wikipedia.org/wiki/Actor_%28programming_language%29" \o "Actor (programming language))**
* **[Ada 95](https://en.wikipedia.org/wiki/Ada_%28programming_language%29" \o "Ada (programming language)) and [Ada 2005](https://en.wikipedia.org/wiki/Ada_%28programming_language%29" \o "Ada (programming language)) (multi-purpose language)**
* **[BETA](https://en.wikipedia.org/wiki/BETA_%28programming_language%29" \o "BETA (programming language))**
* **[Blue](https://en.wikipedia.org/wiki/Blue_%28programming_language%29" \o "Blue (programming language))**
* **[C++](https://en.wikipedia.org/wiki/C%2B%2B" \o "C++)**
* **[C#](https://en.wikipedia.org/wiki/C_Sharp_%28programming_language%29" \o "C Sharp (programming language))**
* **[Ceylon](https://en.wikipedia.org/wiki/Ceylon_%28programming_language%29" \o "Ceylon (programming language))**
* **[Oxygene](https://en.wikipedia.org/wiki/Oxygene_%28programming_language%29" \o "Oxygene (programming language)) (formerly known as Chrome)**
* **[ChucK](https://en.wikipedia.org/wiki/ChucK" \o "ChucK)**
* **[Cobra](https://en.wikipedia.org/wiki/Cobra_%28programming_language%29" \o "Cobra (programming language))**
* **[ColdFusion](https://en.wikipedia.org/wiki/ColdFusion" \o "ColdFusion)**
* **[Curl](https://en.wikipedia.org/wiki/Curl_%28programming_language%29" \o "Curl (programming language))**
* **[D](https://en.wikipedia.org/wiki/D_%28programming_language%29" \o "D (programming language))**
* **[DASL](https://en.wikipedia.org/wiki/Distributed_Application_Specification_Language" \o "Distributed Application Specification Language)**
* **[Delphi](https://en.wikipedia.org/wiki/Object_Pascal" \o "Object Pascal)**
* **[E](https://en.wikipedia.org/wiki/E_%28programming_language%29" \o "E (programming language))**
* **[GNU E](https://en.wikipedia.org/wiki/GNU_E" \o "GNU E)**
* **[eC](https://en.wikipedia.org/wiki/EC_%28programming_language%29" \o "EC (programming language))**
* **[Eiffel](https://en.wikipedia.org/wiki/Eiffel_%28programming_language%29" \o "Eiffel (programming language))** 
  + **[Sather](https://en.wikipedia.org/wiki/Sather" \o "Sather)**
  + **[Ubercode](https://en.wikipedia.org/wiki/Ubercode" \o "Ubercode)**
* **[F-Script](https://en.wikipedia.org/wiki/F-Script_%28programming_language%29" \o "F-Script (programming language))**
* **[Fortran 2003](https://en.wikipedia.org/wiki/Fortran_2003" \o "Fortran 2003)**
* **[Fortress](https://en.wikipedia.org/wiki/Fortress_%28programming_language%29" \o "Fortress (programming language))**
* **[FPr](https://en.wikipedia.org/wiki/FPr_%28programming_language%29" \o "FPr (programming language))**

**28. Object-oriented prototype-based languages**

**[Prototype-based languages](https://en.wikipedia.org/wiki/Prototype-based_programming" \o "Prototype-based programming) are object-oriented languages where the distinction between classes and instances has been removed:**

* **[ABCL/1](https://en.wikipedia.org/wiki/Actor-Based_Concurrent_Language" \o "Actor-Based Concurrent Language)**
* **[ABCL/R](https://en.wikipedia.org/wiki/Actor-Based_Concurrent_Language" \o "Actor-Based Concurrent Language)**
* **[ABCL/R2](https://en.wikipedia.org/wiki/Actor-Based_Concurrent_Language" \o "Actor-Based Concurrent Language)**
* **[ABCL/c plus](https://en.wikipedia.org/wiki/Actor-Based_Concurrent_Language" \o "Actor-Based Concurrent Language)**
* **[Agora](https://en.wikipedia.org/wiki/Agora_%28programming_language%29" \o "Agora (programming language))**
* **[Cecil](https://en.wikipedia.org/wiki/Cecil_%28programming_language%29" \o "Cecil (programming language))**
* **[ECMAScript](https://en.wikipedia.org/wiki/ECMAScript" \o "ECMAScript)** 
  + **[ActionScript](https://en.wikipedia.org/wiki/ActionScript" \o "ActionScript)**
  + **[ECMAScript for XML](https://en.wikipedia.org/wiki/ECMAScript_for_XML" \o "ECMAScript for XML)**
  + **[JavaScript](https://en.wikipedia.org/wiki/JavaScript" \o "JavaScript) (first named Mocha, then LiveScript)**
  + **[JScript](https://en.wikipedia.org/wiki/JScript" \o "JScript)**
* **[Etoys](https://en.wikipedia.org/wiki/Etoys_%28programming_language%29" \o "Etoys (programming language)) in [Squeak](https://en.wikipedia.org/wiki/Squeak" \o "Squeak)**
* **[Io](https://en.wikipedia.org/wiki/Io_%28programming_language%29" \o "Io (programming language))**
* **[Lisaac](https://en.wikipedia.org/wiki/Lisaac" \o "Lisaac)**
* **[Lua](https://en.wikipedia.org/wiki/Lua_%28programming_language%29" \o "Lua (programming language))**
* **[MOO](https://en.wikipedia.org/wiki/MOO_%28programming_language%29" \o "MOO (programming language))**
* **[NewtonScript](https://en.wikipedia.org/wiki/NewtonScript" \o "NewtonScript)**
* **[Obliq](https://en.wikipedia.org/wiki/Obliq" \o "Obliq)**
* **[R](https://en.wikipedia.org/wiki/R_%28programming_language%29" \o "R (programming language))**
* **[REBOL](https://en.wikipedia.org/wiki/REBOL" \o "REBOL)**
* **[Self](https://en.wikipedia.org/wiki/Self_%28programming_language%29" \o "Self (programming language)) (the first prototype-based language, derived from [Smalltalk](https://en.wikipedia.org/wiki/Smalltalk" \o "Smalltalk))**
* **[TADS](https://en.wikipedia.org/wiki/TADS" \o "TADS)**

**29. Off-side rule languages**

**[Off-side rule](https://en.wikipedia.org/wiki/Off-side_rule" \o "Off-side rule) languages are those where blocks are formed, indicated, by their indentation.**

* **[ISWIM](https://en.wikipedia.org/wiki/ISWIM" \o "ISWIM), the abstract language that introduced the rule**
* **[ABC](https://en.wikipedia.org/wiki/ABC_%28programming_language%29" \o "ABC (programming language)), Python's parent** 
  + **[Python](https://en.wikipedia.org/wiki/Python_%28programming_language%29" \o "Python (programming language))** 
    - **[Cobra](https://en.wikipedia.org/wiki/Cobra_%28programming_language%29" \o "Cobra (programming language))**
    - **[Boo](https://en.wikipedia.org/wiki/Boo_%28programming_language%29" \o "Boo (programming language))**
    - **[Genie](https://en.wikipedia.org/wiki/Genie_%28programming_language%29" \o "Genie (programming language))**
* **[Miranda](https://en.wikipedia.org/wiki/Miranda_%28programming_language%29" \o "Miranda (programming language)), Haskell's parent** 
  + **[Orwell](https://en.wikipedia.org/wiki/Orwell_%28programming_language%29" \o "Orwell (programming language))**
  + **[Haskell](https://en.wikipedia.org/wiki/Haskell_%28programming_language%29" \o "Haskell (programming language))** 
    - **[Curry](https://en.wikipedia.org/wiki/Curry_%28programming_language%29" \o "Curry (programming language))**
* **[Elixir](https://en.wikipedia.org/wiki/Elixir_%28programming_language%29" \o "Elixir (programming language))**
* **[F#](https://en.wikipedia.org/wiki/F_Sharp_%28programming_language%29" \o "F Sharp (programming language))**
* **[Nim](https://en.wikipedia.org/wiki/Nim_%28programming_language%29" \o "Nim (programming language))**
* **[Occam](https://en.wikipedia.org/wiki/Occam_%28programming_language%29" \o "Occam (programming language))**
* **[SPIN](https://en.wikipedia.org/wiki/Parallax_Propeller" \o "Parallax Propeller)**
* **[XL](https://en.wikipedia.org/wiki/XL_%28programming_language%29" \o "XL (programming language))**

**30. Procedural languages**

* **[Ada](https://en.wikipedia.org/wiki/Ada_%28programming_language%29" \o "Ada (programming language)) (multi-purpose language)**
* **[ALGOL](https://en.wikipedia.org/wiki/ALGOL" \o "ALGOL) (extremely influential language design – the second high level language compiler)** 
  + **[SMALL Machine Algol Like Language](https://en.wikipedia.org/wiki/SMALL" \o "SMALL)**
* **[Alma-0](https://en.wikipedia.org/wiki/Alma-0" \o "Alma-0)**
* **[BASIC](https://en.wikipedia.org/wiki/BASIC" \o "BASIC) (BASICs are innocent of most modularity in (especially) versions before about 1990)**
* **[BCPL](https://en.wikipedia.org/wiki/BCPL" \o "BCPL)**
* **[BLISS](https://en.wikipedia.org/wiki/BLISS_%28programming_language%29" \o "BLISS (programming language))**
* **[Blue](https://en.wikipedia.org/wiki/Blue_%28programming_language%29" \o "Blue (programming language))**
* **[C](https://en.wikipedia.org/wiki/C_%28programming_language%29" \o "C (programming language))**
* **[C++](https://en.wikipedia.org/wiki/C%2B%2B" \o "C++) (C with objects plus much else, such as, generics through STL)**
* **[Occam](https://en.wikipedia.org/wiki/Occam_%28programming_language%29" \o "Occam (programming language))**
* **[Oriel](https://en.wikipedia.org/wiki/Oriel_%28scripting_language%29" \o "Oriel (scripting language))**
* **[Pascal](https://en.wikipedia.org/wiki/Pascal_%28programming_language%29" \o "Pascal (programming language)) (successor to ALGOL 60, predecessor of Modula-2)** 
  + **[Free Pascal](https://en.wikipedia.org/wiki/Free_Pascal" \o "Free Pascal) (FPC)**
  + **[Object Pascal](https://en.wikipedia.org/wiki/Object_Pascal" \o "Object Pascal) (Delphi)**
* **[PCASTL](https://en.wikipedia.org/wiki/PCASTL" \o "PCASTL)**
* **[Perl](https://en.wikipedia.org/wiki/Perl" \o "Perl)**
* **[Pike](https://en.wikipedia.org/wiki/Pike_%28programming_language%29" \o "Pike (programming language))**
* **[VBScript](https://en.wikipedia.org/wiki/VBScript" \o "VBScript)**
* **[Visual Basic](https://en.wikipedia.org/wiki/Visual_Basic" \o "Visual Basic)**
* **[Visual FoxPro](https://en.wikipedia.org/wiki/Visual_FoxPro" \o "Visual FoxPro)**
* **[Wolfram Language](https://en.wikipedia.org/wiki/Wolfram_Language" \o "Wolfram Language)**
* **[X++](https://en.wikipedia.org/wiki/Microsoft_Dynamics_AX" \o "Microsoft Dynamics AX)**
* **[X#](https://en.wikipedia.org/wiki/X_Sharp_%28programming_language%29" \o "X Sharp (programming language))**
* **[XL](https://en.wikipedia.org/wiki/XL_%28programming_language%29" \o "XL (programming language))**

**31. Reflective languages**

* **[Befunge](https://en.wikipedia.org/wiki/Befunge" \o "Befunge)**
* **[C#](https://en.wikipedia.org/wiki/C_Sharp_%28programming_language%29" \o "C Sharp (programming language))**
* **[Ceylon](https://en.wikipedia.org/wiki/Ceylon_%28programming_language%29" \o "Ceylon (programming language))**
* **[Charm](https://en.wikipedia.org/wiki/Charm_%28language%29" \o "Charm (language))**
* **[ChucK](https://en.wikipedia.org/wiki/ChucK" \o "ChucK)**
* **[Cobra](https://en.wikipedia.org/wiki/Cobra_%28programming_language%29" \o "Cobra (programming language))**
* **[Component Pascal](https://en.wikipedia.org/wiki/Component_Pascal" \o "Component Pascal) [BlackBox Component Builder](https://en.wikipedia.org/wiki/BlackBox_Component_Builder" \o "BlackBox Component Builder)**
* **[Curl](https://en.wikipedia.org/wiki/Curl_%28programming_language%29" \o "Curl (programming language))**
* **[Delphi](https://en.wikipedia.org/wiki/Object_Pascal" \o "Object Pascal)**
* **[eC](https://en.wikipedia.org/wiki/EC_%28programming_language%29" \o "EC (programming language))**
* **[ECMAScript](https://en.wikipedia.org/wiki/ECMAScript" \o "ECMAScript)** 
  + **[ActionScript](https://en.wikipedia.org/wiki/ActionScript" \o "ActionScript)**
  + **[ECMAScript for XML](https://en.wikipedia.org/wiki/ECMAScript_for_XML" \o "ECMAScript for XML)**
  + **[JavaScript](https://en.wikipedia.org/wiki/JavaScript" \o "JavaScript)**
  + **[JScript](https://en.wikipedia.org/wiki/JScript" \o "JScript)**
* **[Eiffel](https://en.wikipedia.org/wiki/Eiffel_%28programming_language%29" \o "Eiffel (programming language))**
* **[Forth](https://en.wikipedia.org/wiki/Forth_%28programming_language%29" \o "Forth (programming language))**
* **[Harbour](https://en.wikipedia.org/wiki/Harbour_%28software%29" \o "Harbour (software))**
* **[Java](https://en.wikipedia.org/wiki/Java_%28programming_language%29" \o "Java (programming language))** 
  + **[Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine" \o "Java virtual machine)**
  + **[Groovy](https://en.wikipedia.org/wiki/Groovy_%28programming_language%29" \o "Groovy (programming language))**
  + **[Join Java](https://en.wikipedia.org/wiki/Join_Java" \o "Join Java)**
  + **[X10](https://en.wikipedia.org/wiki/X10_%28programming_language%29" \o "X10 (programming language))**
* **[Julia](https://en.wikipedia.org/wiki/Julia_%28programming_language%29" \o "Julia (programming language))**
* **[Lisp](https://en.wikipedia.org/wiki/Lisp_%28programming_language%29" \o "Lisp (programming language))** 
  + **[Clojure](https://en.wikipedia.org/wiki/Clojure" \o "Clojure)**
  + **[Common Lisp](https://en.wikipedia.org/wiki/Common_Lisp" \o "Common Lisp)**
  + **[Dylan](https://en.wikipedia.org/wiki/Dylan_%28programming_language%29" \o "Dylan (programming language))**
  + **[Logo](https://en.wikipedia.org/wiki/Logo_%28programming_language%29" \o "Logo (programming language))**
  + **[Scheme](https://en.wikipedia.org/wiki/Scheme_%28programming_language%29" \o "Scheme (programming language))**
* **[Lua](https://en.wikipedia.org/wiki/Lua_%28programming_language%29" \o "Lua (programming language))**
* **[Maude system](https://en.wikipedia.org/wiki/Maude_system" \o "Maude system)**
* **[.NET Framework](https://en.wikipedia.org/wiki/.NET_Framework" \o ".NET Framework) [Common Language Runtime](https://en.wikipedia.org/wiki/Common_Language_Runtime" \o "Common Language Runtime)**

**32. Rule-based languages**

* **[awk](https://en.wikipedia.org/wiki/Awk" \o "Awk)**
* **[CLIPS](https://en.wikipedia.org/wiki/CLIPS" \o "CLIPS)**
* **[Constraint Handling Rules](https://en.wikipedia.org/wiki/Constraint_Handling_Rules" \o "Constraint Handling Rules)**
* **[Drools](https://en.wikipedia.org/wiki/Drools" \o "Drools)**
* **[GOAL agent programming language](https://en.wikipedia.org/wiki/GOAL_agent_programming_language" \o "GOAL agent programming language)**
* **[Jess](https://en.wikipedia.org/wiki/Jess_%28programming_language%29" \o "Jess (programming language))**
* **[make](https://en.wikipedia.org/wiki/Make" \o "Make)**
* **[OPS5](https://en.wikipedia.org/wiki/OPS5" \o "OPS5)**
* **[Prolog](https://en.wikipedia.org/wiki/Prolog" \o "Prolog)**
* **[ToonTalk](https://en.wikipedia.org/wiki/ToonTalk" \o "ToonTalk) – robots are rules**
* **[Mathematica](https://en.wikipedia.org/wiki/Mathematica" \o "Mathematica)**
* **[XSLT](https://en.wikipedia.org/wiki/XSLT" \o "XSLT)[[citation needed](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed" \o "Wikipedia:Citation needed)]**
* **[Wolfram Language](https://en.wikipedia.org/wiki/Wolfram_Language" \o "Wolfram Language)**

**33. Scripting languages**

* **[AppleScript](https://en.wikipedia.org/wiki/AppleScript" \o "AppleScript)**
* **[AWK](https://en.wikipedia.org/wiki/AWK" \o "AWK)**
* **[BeanShell](https://en.wikipedia.org/wiki/BeanShell" \o "BeanShell)**
* **[Bash](https://en.wikipedia.org/wiki/Bash_%28Unix_shell%29" \o "Bash (Unix shell))**
* **[Ch](https://en.wikipedia.org/wiki/Ch_%28computer_programming%29" \o "Ch (computer programming)) (Embeddable C/C++ interpreter)**
* **[CLIST](https://en.wikipedia.org/wiki/CLIST" \o "CLIST)**
* **[ColdFusion](https://en.wikipedia.org/wiki/ColdFusion" \o "ColdFusion)**
* **[ECMAScript](https://en.wikipedia.org/wiki/ECMAScript" \o "ECMAScript)** 
  + **[ActionScript](https://en.wikipedia.org/wiki/ActionScript" \o "ActionScript)**
  + **[ECMAScript for XML](https://en.wikipedia.org/wiki/ECMAScript_for_XML" \o "ECMAScript for XML)**
  + **[JavaScript](https://en.wikipedia.org/wiki/JavaScript" \o "JavaScript) (first named Mocha, then LiveScript)**
  + **[JScript](https://en.wikipedia.org/wiki/JScript" \o "JScript)**
* **[CMS EXEC](https://en.wikipedia.org/wiki/CMS_EXEC" \o "CMS EXEC)**
* **[EXEC 2](https://en.wikipedia.org/wiki/EXEC_2" \o "EXEC 2)**

**34. Stack-based languages**

* **[Beatnik](https://en.wikipedia.org/wiki/Beatnik_%28programming_language%29" \o "Beatnik (programming language))**
* **[colorForth](https://en.wikipedia.org/wiki/ColorForth" \o "ColorForth)**
* **[Factor](https://en.wikipedia.org/wiki/Factor_%28programming_language%29" \o "Factor (programming language))**
* **[Forth](https://en.wikipedia.org/wiki/Forth_%28programming_language%29" \o "Forth (programming language))**
* **[Joy](https://en.wikipedia.org/wiki/Joy_%28programming_language%29" \o "Joy (programming language)) (all functions work on parameter stacks instead of named parameters)**
* **[Piet](https://en.wikipedia.org/wiki/Piet_%28programming_language%29" \o "Piet (programming language))**
* **[Poplog](https://en.wikipedia.org/wiki/Poplog" \o "Poplog) via its implementation language [POP-11](https://en.wikipedia.org/wiki/POP-11" \o "POP-11)**
* **[PostScript](https://en.wikipedia.org/wiki/PostScript" \o "PostScript)**
* **[RPL](https://en.wikipedia.org/wiki/RPL_%28programming_language%29" \o "RPL (programming language))**

**35. Synchronous languages**

* **[Argus](https://en.wikipedia.org/wiki/Argus_%28programming_language%29" \o "Argus (programming language))**
* **[Averest](https://en.wikipedia.org/wiki/Averest" \o "Averest)**
* **[Esterel](https://en.wikipedia.org/wiki/Esterel" \o "Esterel)**
* **[Lustre](https://en.wikipedia.org/wiki/Lustre_%28programming_language%29" \o "Lustre (programming language))**
* **[SyncCharts](https://en.wikipedia.org/wiki/SyncCharts" \o "SyncCharts)**

**36. Syntax handling languages**

**These languages assist with generating [lexical analyzers](https://en.wikipedia.org/wiki/Lexical_analysis" \o "Lexical analysis) and [parsers](https://en.wikipedia.org/wiki/Parsing" \o "Parsing) for [Context-free grammars](https://en.wikipedia.org/wiki/Context-free_grammar" \o "Context-free grammar).**

* **[ANTLR](https://en.wikipedia.org/wiki/ANTLR" \o "ANTLR)**
* **[Coco/R](https://en.wikipedia.org/wiki/Coco/R" \o "Coco/R) (EBNF with semantics)**
* **[GNU bison](https://en.wikipedia.org/wiki/GNU_bison" \o "GNU bison) (FSF's version of Yacc)**
* **GNU [Flex](https://en.wikipedia.org/wiki/Flex_lexical_analyser" \o "Flex lexical analyser) (FSF's version of Lex)**
* **glex/gyacc (GoboSoft compiler compiler to Eiffel)**
* **[lex](https://en.wikipedia.org/wiki/Lex_%28software%29" \o "Lex (software)) (Lexical Analysis, from Bell Labs)**
* **[M4](https://en.wikipedia.org/wiki/M4_%28computer_language%29" \o "M4 (computer language))**
* **[yacc](https://en.wikipedia.org/wiki/Yacc" \o "Yacc) (yet another compiler compiler, from Bell Labs)**
* **[JavaCC](https://en.wikipedia.org/wiki/JavaCC" \o "JavaCC)**

**37. Visual languages**

* **[Analytica](https://en.wikipedia.org/wiki/Analytica_%28software%29" \o "Analytica (software))**
* **[Blockly](https://en.wikipedia.org/wiki/Blockly" \o "Blockly)**
* **[CODE](https://en.wikipedia.org/wiki/CODE_%28programming_language%29" \o "CODE (programming language))**
* **[DRAKON](https://en.wikipedia.org/wiki/DRAKON" \o "DRAKON)**
* **[Fabrik](https://en.wikipedia.org/wiki/Fabrik_%28software%29" \o "Fabrik (software))**
* **G (used in [LabVIEW](https://en.wikipedia.org/wiki/LabVIEW" \o "LabVIEW))**
* **[Lava](https://en.wikipedia.org/wiki/Lava_%28programming_language%29" \o "Lava (programming language))**
* **[Limnor](https://en.wikipedia.org/wiki/Limnor" \o "Limnor)**
* **[Max](https://en.wikipedia.org/wiki/Max_%28software%29" \o "Max (software))**
* **[NXT-G](https://en.wikipedia.org/wiki/NXT-G" \o "NXT-G)**
* **[Pict programming language](https://en.wikipedia.org/wiki/Pict_%28programming_language%29" \o "Pict (programming language))**
* **[Prograph](https://en.wikipedia.org/wiki/Prograph" \o "Prograph)**
* **[Pure Data](https://en.wikipedia.org/wiki/Pure_Data" \o "Pure Data)**

**38. Wirth languages**

* **[ALGOL W](https://en.wikipedia.org/wiki/ALGOL_W" \o "ALGOL W)**
* **[Euler](https://en.wikipedia.org/wiki/Euler_%28programming_language%29" \o "Euler (programming language))**
* **[Modula](https://en.wikipedia.org/wiki/Modula" \o "Modula)**
* **[Modula-2](https://en.wikipedia.org/wiki/Modula-2" \o "Modula-2) (and Modula 3, etc. variants)** 
  + **[Obliq](https://en.wikipedia.org/wiki/Obliq" \o "Obliq) Modula 3 variant**
* **[Oberon](https://en.wikipedia.org/wiki/Oberon_%28programming_language%29" \o "Oberon (programming language)) (Oberon, Oberon-07, and Oberon-2)** 
  + **[Component Pascal](https://en.wikipedia.org/wiki/Component_Pascal" \o "Component Pascal)**
  + **[Lagoona](https://en.wikipedia.org/wiki/Lagoona_%28programming_language%29" \o "Lagoona (programming language))**
  + **[Oberon-2](https://en.wikipedia.org/wiki/Oberon-2_%28programming_language%29" \o "Oberon-2 (programming language))**
* **[Pascal](https://en.wikipedia.org/wiki/Pascal_%28programming_language%29" \o "Pascal (programming language))** 
  + **[Object Pascal](https://en.wikipedia.org/wiki/Object_Pascal" \o "Object Pascal) ("umbrella" name for [Delphi](https://en.wikipedia.org/wiki/Embarcadero_Delphi" \o "Embarcadero Delphi), [Free Pascal](https://en.wikipedia.org/wiki/Free_Pascal" \o "Free Pascal), [Oxygene](https://en.wikipedia.org/wiki/Oxygene_%28programming_language%29" \o "Oxygene (programming language)) and others)**

**39. XML-based languages**

**These are languages based on or that operate on [XML](https://en.wikipedia.org/wiki/XML" \o "XML).**

* **[Ant](https://en.wikipedia.org/wiki/Apache_Ant" \o "Apache Ant)**
* **[Cω](https://en.wikipedia.org/wiki/C%CF%89" \o "Cω)**
* **[ECMAScript for XML](https://en.wikipedia.org/wiki/ECMAScript_for_XML" \o "ECMAScript for XML)**
* **[MXML](https://en.wikipedia.org/wiki/MXML" \o "MXML)**
* **[LZX](https://en.wikipedia.org/wiki/OpenLaszlo" \o "OpenLaszlo)**
* **[XAML](https://en.wikipedia.org/wiki/XAML" \o "XAML)**
* **[XPath](https://en.wikipedia.org/wiki/XPath" \o "XPath)**
* **[XQuery](https://en.wikipedia.org/wiki/XQuery" \o "XQuery)**
* **[XProc](https://en.wikipedia.org/wiki/XProc" \o "XProc)**
* **[XSLT](https://en.wikipedia.org/wiki/XSL_Transformations" \o "XSL Transformations)**

**Top of Form**

**Bottom of Form**